Ref #	Hits	Search Query	DBs	Default Operator	Plurals	Time Stamp
L1	206	717/130.ccls.	USPAT	OR	OFF	2005/11/22 12:15
L2	50	717/133.ccls.	USPAT	OR	OFF	2005/11/22 12:15
L3	311	717/127.ccls.	USPAT	OR	OFF	2005/11/22 12:15
L4	200	714/35.ccls.	USPAT	OR	OFF	2005/11/22 12:15
L5	0	717/130.ccls. and ("yield point" or (safe$2 adj point)) and (condition$4 or uncondition$4)	USPAT	OR	OFF	2005/11/22 12:19
L6	0	717/133.ccls. and ("yield point" or (safe$2 adj point)) and (condition$4 or uncondition$4)	USPAT	OR	OFF	2005/11/22 12:19
L7	0	717/127.ccls. and ("yield point" or (safe$2 adj point)) and (condition$4 or uncondition$4)	USPAT	OR	OFF	2005/11/22 12:20
L10	137	714/35.ccls. and (condition$4 or uncondition$4)	USPAT	OR	OFF	2005/11/22 12:20
L11	0	714/35.ccls. and ("yield point" or (safe$2 adj point)) and (condition$4 or uncondition$4)	USPAT	OR	OFF	2005/11/22 12:20
L12	182	717/127.ccls. and (condition$4 or uncondition$4)	USPAT	OR	OFF	2005/11/22 12:20
L13	30	717/133.ccls. and (condition$4 or uncondition$4)	USPAT	OR	OFF	2005/11/22 12:21
L14	130	717/130.ccls. and (condition$4 or uncondition$4)	USPAT	OR	OFF	2005/11/22 12:30
L15	0	"5710724".pn. and condition$4 and uncondition$4	USPAT	OR	OFF	2005/11/22 12:31
L16	1	"5710724".pn. and (condition$4 or uncondition$4)	USPAT	OR	OFF	2005/11/22 12:31

L17	23	("4231106" | "4740895" | "5313616" | "5335344" | "5386522" | "5430878" | "5481688" | "5528753" | "5535329" | "5537541" | "5539907" | "5560013").PN. OR ("5710724").URPN.	US-PGPUB;
			USPAT;
			USOCR
				OR	OFF	2005/11/22 12:31

L18	14	(("4231106" | "4740895" | "5313616" | "5335344" | "5386522" | "5430878" | "5481688" | "5528753" | "5535329" | "5537541" | "5539907" | "5560013").PN. OR ("5710724").URPN.) and (condition$4 or uncondition$4)	US-PGPUB;
			USPAT;
			USOCR
				OR	OFF	2005/11/22 12:37

L19	90	(execut$3 or run$4) near3 ((condition$4 or uncondition$4) near3 instrument$5)	US-PGPUB;
			USPAT;
			USOCR
				OR	OFF	2005/11/22 12:38

S1	3	((insert$3 near3 (handler or interrupt or event)) near5 (sampl$3 or measur$4)) and ((boolean or condition$4 or guard$3 or predicat$3) near3 (sampl$3 or measur$4))	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2005/10/31 15:17
S2	48	((insert$3 near3 (handler or interrupt or event)) same (sampl$3 or measur$4)) and ((boolean or condition$4 or guard$3 or predicat$3) near3 (sampl$3 or measur$4))	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2005/10/31 15:30
S3	22	((instrument$5 near3 (handler or interrupt or event)) same (sampl$3 or measur$4)) and ((boolean or condition$4 or guard$3 or predicat$3) near3 (sampl$3 or measur$4))	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2005/10/31 15:30
S4	57	("5151981").URPN.	USPAT	OR	OFF	2005/10/31 15:56
S5	12	("6237073").URPN.	USPAT	OR	OFF	2005/10/31 16:45
S6	10	("6237073").URPN. and (condition$3 or boolean or predicate)	USPAT	OR	OFF	2005/10/31 16:46
S7	10	("6237073").URPN. and (condition$3 or boolean or predicate)	USPAT	OR	OFF	2005/10/31 16:56
S8	381050	(sampl$3 or profil$3 or monitor$3 or measur$4) same (condition$3 or boolean or predicate)	USPAT	OR	OFF	2005/10/31 16:57
S9	23874	(insert$3 or instrument$5) same (sampl$3 or profil$3 or monitor$3 or measur$4) same (condition$3 or boolean or predicate)	USPAT	OR	OFF	2005/10/31 16:57
S10	4126	(insert$3 or instrument$5) same (sampl$3 or profil$3 or monitor$3 or measur$4) near3 (condition$3 or boolean or predicate)	USPAT	OR	OFF	2005/10/31 16:58
S11	2988	(insert$3 or instrument$5) same (sampl$3 or profil$3 or monitor$3 or measur$4) near2 (condition$3 or boolean or predicate)	USPAT	OR	OFF	2005/10/31 16:58
S12	724	(insert$3 or instrument$5) near5 ((sampl$3 or profil$3 or monitor$3 or measur$4) near2 (condition$3 or boolean or predicate))	USPAT	OR	OFF	2005/10/31 16:59
S13	3	(insert$3 or instrument$5) near5 ((sampl$3 or profil$3 or monitor$3 or measur$4) near2 (condition$3 or boolean or predicate)) and 717/???.ccls.	USPAT	OR	OFF	2005/10/31 17:00

S14	21	(insert$3 or instrument$5) near5 execut$4 and ((sampl$3 or profil$3 or monitor$3 or measur$4) near2 (condition$3 or boolean or predicate)) and (handler or function or operation or call) and 717/???.ccls.	USPAT	OR	OFF	2005/10/31 17:01
S15	14	("5960198").URPN.	USPAT	OR	OFF	2005/11/01 07:13
S16	4	("5710724" "6249912" "6327699" "5995754").pn.	USPAT	OR	OFF	2005/11/01 07:38
S17	1	"5960198".pn.	USPAT	OR	OFF	2005/11/01 07:38
S18	14	("5960198").URPN.	USPAT	OR	OFF	2005/11/01 07:39
S19	0	"6971091".pn.	USPAT	OR	OFF	2005/11/14 15:56
S20	0	"adaptively optimizing program".ti.	USPAT	OR	OFF	2005/11/14 15:57
S21	1	"international business machines". as. and arnold.in. and fink.in.	USPAT	OR	OFF	2005/11/14 16:02
S22	28	"international business machines". as. and "yield point"	USPAT	OR	OFF	2005/11/14 16:13
S23	0	"sampling at selected program points"	USPAT	OR	OFF	2005/11/14 16:13
S24	31	"sampling" and "program points"	USPAT	OR	OFF	2005/11/14 16:13

Google Scholar BETA

conditional sampling yield point | 1985 - 2000 | Search

Ac
Sc
Sc

Scholar Results **1** - **10** of about **6,390** for **conditional sampling yield point**. (**0.14** seconds)

Conditional-sampling spectrograph detection system for fluorescence measurements of individual ...
P Nachman, G Chen, RG Pinnick, SC Hill, RK Chang, ... - Applied Optics, 1996 - ao.osa.org
... circulating laser beam at the **point** within the ... of the flow rate might **yield** operating conditions ... 6. 1a2 Fluorescence spectra taken with **conditional sampling**. ...
Cited by 10 - Web Search - aoot.osa.org - adsabs.harvard.edu - csa.com - all 7 versions »

Size-biased **sampling** of Poisson **point** processes and excursions
M Perman, J Pitman, M Yor - Probability Theory and Related Fields, 1992 - springerlink.com
... d. derived from A(.)by deletion of the **point** in dx ... A(0 e dx is chosen by size-biased **sampling** from A ... In particu- lar, the **conditional** probability for 2) given 1 ...
Cited by 51 - Web Search - emis.de - emis.de

Convergence and accuracy of Gibbs **sampling** for **conditional** distributions in generalized linear ...
JE Kolassa - Ann. Statist, 1999 - projecteuclid.org
... convergence conditions for a Markov chain con- structed using Gibbs **sampling**, when the equilibrium distribution is the **conditional sampling** distribution of ...
Cited by 3 - Web Search - projecteuclid.org

Change-**point** problems for the von Mises distribution
K GHOSH, SR JAMMALAMADAKA, M VASUDAVEN - Journal of Applied Statistics, 1999 - dx.doi.org
... K > c (6) where the cut-oþ **point** c is determined based on the signi® cance level of the test. ... j , **conditional** on R. However, the joint distributions of (R
Web Search - taylorandfrancis.metapress.com - ingentaconnect.com - ideas.repec.org - all 5 versions »

A Bayesian Growth and **Yield** Model for Slash Pine Plantations
E Green, W Strawderman - Journal of Applied Statistics, 1996 - dx.doi.org
... j trees to **yield** an estimate of the ... For examples of Gibbs **sampling** in forestry problems, see ... sampler, one must derive the `full **conditional**' distribution for ...
Cited by 16 - Web Search - taylorandfrancis.metapress.com - ingentaconnect.com - ideas.repec.org

Some observations on fitting assumed diameter distributions to horizontal **point sampling** data
JH Gove - Can. J. For. Res, 2000 - article.pubs.nrc-cnrc.gc.ca
... or lack thereof) in these estimates to **sampling** intensity per **point** ... In general, larger sample sizes per **point** are required to **yield** reliable parameter ...
Cited by 4 - Web Search - article.pubs.nrc-cnrc.gc.ca

Application of Gibbs **sampling** for inference in a mixed major gene-polygenic inheritance model in ...
LLG Janss, R Thompson, AM Arendonk - Theoretical and Applied Genetics, 1995 - springerlink.com
... In the reduced **conditional** density for a sire, the ... Further nota- tion is as in **sampling** steps $2.1 ... or different functions of parameters may **yield** different K's ...
Cited by 71 - Web Search

Detection of turbulent coherent motions in a forest canopy part II: Time-scales and **conditional** ...

S Collineau, Y Brunet - Boundary-Layer Meteorology, 1993 - springerlink.com
... the possibility of using wavelets to perform **conditional sampling**. ... **Conditional** averaging will then be performed on ... most cases coherent motions **yield** ramps in ...
Cited by 27 - Web Search

Nulling' filters and the separation of transparent motions
T Darrell, E Simoncelli - 1993 - ieeexplore.ieee.org
... sufficiently smooth that a relatively coarse **sampling** of the ... be the maximum value of this **conditional** probability over ... that **point**) will al- ways **yield** a lower ...
Cited by 12 - Web Search - csa.com

Comparisons among some estimators in misspecified linear models with multicollinearity
N Sarkar - Annals of the Institute of Statistical Mathematics, 1989 - springerlink.com
... of the predictor of the **conditional** mean of ... of least squares produces large **sampling** variances of ... These often **yield point** estimates (of parameters) superior to ...
Cited by 3 - Web Search

Gooooooooogle ►
Result Page: 1 2 3 4 5 6 7 8 9 10 **Next**

conditional sampling yield point [Search]

Google Home - About Google - About Google Scholar

©2005 Google

Google Scholar BETA conditional sampling yield point unconditonally 1985 - 2000 Search Ac
Sc
Sc

Your search - **conditional sampling yield point unconditonally** - did not match any articles published between **1985** and **2000**.

Suggestions:

- Make sure all words are spelled correctly.
- Try different keywords.
- Try more general keywords.
- Try fewer keywords.
- Try searching over a larger date range.
- Try your query on the entire web.

Google Home - About Google - About Google Scholar

PORTAL
USPTO

Subscribe (Full Service) Register (Limited Service, Free) Lo

Search: ● The ACM Digital Library ○ The Guide

+conditional* +unconditional* +sampling +instrumentation

THE ACM DIGITAL LIBRARY

Feedback Report a problem Satisfaction su

Published since January 1985 and Published before November 2000
Terms used
conditional unconditional sampling instrumentation

Found **37** of **83**

Sort results by: relevance

Display results: expanded form

Save results to a Binder

Search Tips

□ Open results in a new window

Try an Advanced Search
Try this search in The ACM Guide

Results 1 - 20 of 37

Result page: **1** 2 next

Relevance scale

1 A software instruction counter
J. M. Mellor-Crummey, T. J. LeBlanc
April 1989 **ACM SIGARCH Computer Architecture News , Proceedings of the third international conference on Architectural support for programming languages and operating systems ASPLOS-III**, Volume 17 Issue 2
Publisher: ACM Press
Full text available: pdf(997.70 KB) Additional Information: full citation, abstract, references, citings, index terms

Although several recent papers have proposed architectural support for program debugging and profiling, most processors do not yet provide even basic facilities, such as an instruction counter. As a result, system developers have been forced to invent software solutions. This paper describes our implementation of a software instruction counter for program debugging. We show that an instruction counter can be reasonably implemented in software, often with less than 10% execution overhead. Ou ...

2 Optimally profiling and tracing programs
Thomas Ball, James R. Larus
July 1994 **ACM Transactions on Programming Languages and Systems (TOPLAS)**, Volume 16 Issue 4
Publisher: ACM Press
Full text available: pdf(2.84 MB) Additional Information: full citation, abstract, references, citings, index terms, review

This paper describes algorithms for inserting monitoring code to profile and trace programs. These algorithms greatly reduce the cost of measuring programs with respect to the commonly used technique of placing code in each basic block. Program profiling counts the number of times each basic block in a program executes. Instruction tracing records the sequence of basic blocks traversed in a program execution. The algorithms optimize the placement of counting/tracing code with respect to the ...

Keywords: control-flow graph, instruction tracing, instrumentation, profiling

3 Reducing indirect function call overhead in C++ programs
Brad Calder, Dirk Grunwald
February 1994 **Proceedings of the 21st ACM SIGPLAN-SIGACT symposium on Principles of programming languages**
Publisher: ACM Press
Full text available: pdf(1.27 MB) Additional Information: full citation, abstract, references, citings, index terms

Modern computer architectures increasingly depend on mechanisms that estimate future control flow decisions to increase performance. Mechanisms such as speculative execution and prefetching are becoming standard architectural mechanisms that rely on control flow prediction to prefetch and speculatively execute future instructions. At the same time, computer programmers are increasingly turning to object-oriented languages to increase their ...

Keywords: customization, object oriented programming, optimization, profile-based optimization

4 Experience with a software-defined machine architecture
David W. Wall
May 1992 **ACM Transactions on Programming Languages and Systems (TOPLAS),** Volume 14 Issue 3
Publisher: ACM Press
Full text available: pdf(2.86 MB) Additional Information: full citation, abstract, references, citings, index terms, review

We have built a system in which the compiler back end and the linker work together to present an abstract machine at a considerably higher level than the actual machine. The intermediate language translated by the back end is the target language of all high-level compilers and is also the only assembly language generally available. This lets us do intermodule register allocation, which would be harder if some of the code in the program had come from a traditional assembler, out of sight of ...

Keywords: RISC, graph coloring, intermediate language, interprocedural, optimization, pipeline scheduling, profiling, register allocation, register windows

5 Predicting conditional branch directions from previous runs of a program
Joseph A. Fisher, Stefan M. Freudenberger
September 1992 **ACM SIGPLAN Notices , Proceedings of the fifth international conference on Architectural support for programming languages and operating systems ASPLOS-V,** Volume 27 Issue 9
Publisher: ACM Press
Full text available: pdf(1.13 MB) Additional Information: full citation, references, citings, index terms

6 Static correlated branch prediction
Cliff Young, Michael D. Smith
September 1999 **ACM Transactions on Programming Languages and Systems (TOPLAS),** Volume 21 Issue 5
Publisher: ACM Press
Full text available: pdf(508.49 KB) Additional Information: full citation, abstract, references, citings, index terms

Recent work in history-based branch prediction uses novel hardware structures to capture branch correlation and increase branch prediction accuracy. Branch correlation occurs when the outcome of a conditional branch can be accurately predicted by observing the outcomes of previously executed branches in the dynamic instruction stream. In this article, we show how to instrument a program so that it is practical to collect run-time statistics that indicate where branch correl ...

Keywords: branch correlation, branch prediction, path profiling, profile-driven optimization

7 System architectures for computer music
John W. Gordon
June 1985 **ACM Computing Surveys (CSUR),** Volume 17 Issue 2
Publisher: ACM Press
Full text available: pdf(4.61 MB) Additional Information: full citation, abstract, references, citings, index terms, review

Computer music is a relatively new field. While a large proportion of the public is aware of computer music in one form or another, there seems to be a need for a better understanding of its capabilities and limitations in terms of synthesis, performance, and recording hardware. This article addresses that need by surveying and discussing the architecture of existing computer music systems. System requirements vary according to what the system will be used for. Common uses for co ...

8 Evidence-based static branch prediction using machine learning
Brad Calder, Dirk Grunwald, Michael Jones, Donald Lindsay, James Martin, Michael Mozer, Benjamin Zorn
January 1997 **ACM Transactions on Programming Languages and Systems (TOPLAS),** Volume 19 Issue 1
Publisher: ACM Press
Full text available: pdf(515.50 KB) Additional Information: full citation, abstract, references, citings, index terms

Correctly predicting the direction that branches will take is increasingly important in today's wide-issue computer architectures. The name program-based branch prediction is given to static branch prediction techniques that base their prediction on a program's structure. In this article, we investigate a new approach to program-based branch prediction that uses a body of existing programs to predict the branch behavior in a new program. We call this approach to program-ba ...

Keywords: branch prediction, decision trees, machine learning, neural networks, performance evaluation, program optimization

9 Generating testing and analysis tools with Aria
Premkumar T. Devanbu, David S. Rosenblum, Alexander L. Wolf
January 1996 **ACM Transactions on Software Engineering and Methodology (TOSEM),** Volume 5 Issue 1
Publisher: ACM Press
Full text available: pdf(1.53 MB) Additional Information: full citation, abstract, references, citings, index terms, review

Many software testing and analysis tools manipulate graph representations of programs, such as abstract syntax trees or abstract semantics graphs. Handcrafting such tools in conventional programming languages can be difficult, error prone, and time consuming. Our approach is to use application generators targeted for the domain of graph-representation-based testing and analysis tools. Moreover, we generate the generators themselves, so that the development of tools based on different langua ...

Keywords: Aria, Genoa, Reprise, application generators, program dependence graphs, program representations, software analysis, software testing, tools

10 Fast and accurate instruction fetch and branch prediction
B. Calder, D. Grunwald
April 1994 **ACM SIGARCH Computer Architecture News , Proceedings of the 21ST annual international symposium on Computer architecture ISCA '94,** Volume 22 Issue 2
Publisher: IEEE Computer Society Press, ACM Press
Full text available: pdf(1.07 MB) Additional Information: full citation, abstract, references, citings, index terms

Accurate branch prediction is critical to performance; mispredicted branches mean that ten's of cycles may be wasted in superscalar architectures. Architectures combining very effective branch prediction mechanisms coupled with modified branch target buffers (BTB's) have been proposed for wide-issue processors. These mechanisms require considerable processor resources Concurrently, the larger address space of 64-bit architectures introduce new obstacles and opportunities. A larger address space ...

11 Variable length path branch prediction
Jared Stark, Marius Evers, Yale N. Patt
October 1998 **ACM SIGPLAN Notices , ACM SIGOPS Operating Systems Review , Proceedings of the eighth international conference on Architectural support for programming languages and operating systems ASPLOS-VIII,** Volume 33 , 32 Issue 11 , 5
Publisher: ACM Press
Full text available: pdf(1.24 MB) Additional Information: full citation, abstract, references, citings, index terms

Accurate branch prediction is required to achieve high performance in deeply pipelined, wide-issue processors. Recent studies have shown that conditional and indirect (or computed) branch targets can be accuratelypredicted by recording the path, which consists of the target addresses of recent branches, leading up to the branch. In current path based branch predictors, the *N* most recent target addresses are hashed together to form an index into a table, where *N* is some fixed intege ...

12 Corpus-based static branch prediction

Brad Calder, Dirk Grunwald, Donald Lindsay, James Martin, Michael Mozer, Benjamin Zorn
June 1995 **ACM SIGPLAN Notices , Proceedings of the ACM SIGPLAN 1995 conference on Programming language design and implementation PLDI '95**, Volume 30 Issue 6
Publisher: ACM Press

Full text available: pdf(1.35 MB) Additional Information: full citation, abstract, references, citings, index terms

Correctly predicting the direction that branches will take is increasingly important in today's wide-issue computer architectures. The name program-based branch prediction is given to static branch prediction techniques that base their prediction on a program's structure. In this paper, we investigate a new approach to program-based branch prediction that uses a body of existing programs to predict the branch behavior in a new program. We call this approach to program-based ...

13 An algorithm for optimal lambda calculus reduction

John Lamping
December 1989 **Proceedings of the 17th ACM SIGPLAN-SIGACT symposium on Principles ol programming languages**
Publisher: ACM Press

Full text available: pdf(1.56 MB) Additional Information: full citation, abstract, references, citings, index terms

We present an algorithm for lambda expression reduction that avoids any copying that could later cause duplication of work. It is optimal in the sense defined by Lévy. The basis of the algorithm is a graphical representation of the kinds of commonality that can arise from substitutions; the idea can be adapted to represent other kinds of expressions besides lambda expressions. The algorithm is also well suited to parallel implementations, consisting of a fixed set of local graph rewr ...

14 Execution characteristics of desktop applications on Windows NT

Dennis C. Lee, Patrick J. Crowley, Jean-Loup Baer, Thomas E. Anderson, Brian N. Bershad
April 1998 **ACM SIGARCH Computer Architecture News , Proceedings of the 25th annual international symposium on Computer architecture ISCA '98**, Volume 26 Issue 3
Publisher: IEEE Computer Society, ACM Press

Full text available: pdf(1.42 MB) Publisher Site Additional Information: full citation, abstract, references, citings, index terms

This paper examines the performance of desktop applications running on the Microsoft Windows NT operating system on Intel x86 processors, and contrasts these applications to the programs in the integer SPEC95 benchmark suite. We present measurements of basic instruction set and program characteristics, and detailed simulation results of the way these programs use the memory system and processor branch architecture. We show that the desktop applications have similar characteristics to the integer ...

15 High-level design verification of microprocessors via error modeling

D. Van Campenhout, H. Al-Asaad, J. P. Hayes, T. Mudge, R. B. Brown
October 1998 **ACM Transactions on Design Automation of Electronic Systems (TODAES)**, Volume 3 Issue 4

Publisher: ACM Press

Full text available: pdf(174.30 KB) Additional Information: full citation, abstract, references, citings, index terms

A design verification methodology for microprocessor hardware based on modeling design errors and generating simulation vectors for the modeled errors via physical fault testing techniques is presented. We have systematically collected design error data from a number of microprocessor design projects. The error data is used to derive error models suitable for design verification testing. A class of basic error models is identified and shown to yield tests that provide good coverage of comm ...

Keywords: design errors, design verification, error modeling

16 Using branch handling hardware to support profile-driven optimization

Thomas M. Conte, Burzin A. Patel, J. Stan Cox

November 1994 **Proceedings of the 27th annual international symposium on Microarchitecture**

Publisher: ACM Press

Full text available: pdf(954.48 KB) Additional Information: full citation, abstract, references, citings, index terms

Profile-based optimizations can be used for instruction scheduling, loop scheduling, data preloading, function in-lining, and instruction cache performance enhancement. However, these techniques have not been embraced by software vendors because programs instrumented for profiling run 2–30 times slower, an awkward compile-run-recompile sequence is required, and a test input suite must be collected and validated for each program. This paper proposes using existing bran ...

17 A scalable cross-platform infrastructure for application performance tuning using hardware counter

S. Browne, J. Dongarra, N. Garner, K. London, P. Mucci

November 2000 **Proceedings of the 2000 ACM/IEEE conference on Supercomputing (CDROM)**

Publisher: IEEE Computer Society

Full text available: pdf(2.82 MB) Publisher Site Additional Information: full citation, abstract, references, citings, index terms

The purpose of the PAPI project is to specify a standard API for accessing hardware performance counters available on most modern microprocessors. These counters exist as a small set of registers that count "events", which are occurrences of specific signals and states related to the processor's function. Monitoring these events facilitates correlation between the structure of source/object code and the efficiency of the mapping of that code to the underlying architecture. This ...

18 Dynamo: a transparent dynamic optimization system

Vasanth Bala, Evelyn Duesterwald, Sanjeev Banerjia

May 2000 **ACM SIGPLAN Notices , Proceedings of the ACM SIGPLAN 2000 conference on Programming language design and implementation PLDI '00**, Volume 35 Issue 5

Publisher: ACM Press

Full text available: pdf(156.03 Additional Information: full citation, abstract, references, KB) citings, index terms

We describe the design and implementation of Dynamo, a software dynamic optimization system that is capable of transparently improving the performance of a native instruction stream as it executes on the processor. The input native instruction stream to Dynamo can be dynamically generated (by a JIT for example), or it can come from the execution of a statically compiled native binary. This paper evaluates the Dynamo system in the latter, more challenging situation, in order to emphasize the ...

19 Instruction cache fetch policies for speculative execution
Dennis Lee, Jean-Loup Baer, Brad Calder, Dirk Grunwald
May 1995 **ACM SIGARCH Computer Architecture News , Proceedings of the 22nd annual international symposium on Computer architecture ISCA '95,** Volume 23 Issue 2
Publisher: ACM Press
Full text available: pdf(1.14 MB) Additional Information: full citation, abstract, references, citings, index terms

Current trends in processor design are pointing to deeper and wider pipelines and superscalar architectures. The efficient use of these resources requires *speculative execution,* a technique whereby the processor continues executing the predicted path of a branch before the branch condition is resolved.In this paper, we investigate the implications of speculative execution on instruction cache performance. We explore policies for managing instruction cache misses ranging from aggressive po ...

20 Combined correlation induction strategies for designed simulation experiments
Chimyung Kwon, Jeffrey D. Tew
December 1993 **Proceedings of the 25th conference on Winter simulation**
Publisher: ACM Press
Full text available: pdf(909.03 KB) Additional Information: full citation, references, citings

Results 1 - 20 of 37 Result page: 1 2 next

Terms of Usage Privacy Policy Code of Ethics Contact Us

Useful downloads: Adobe Acrobat QuickTime Windows Media Player Real Playe

PORTAL
USPTO

Subscribe (Full Service) Register (Limited Service, Free) Lo

Search: ● The ACM Digital Library ○ The Guide

+conditional* +unconditional* +yield +points

THE ACM DIGITAL LIBRARY

Feedback Report a problem Satisfaction su

Published since January 1985 and Published before November 2000

Found **277** of **83**

Terms used **conditional unconditional yield points**

Sort results by: relevance

Display results: expanded form

Save results to a Binder

Search Tips

□ Open results in a new window

Try an Advanced Search

Try this search in The ACM Guide

Results 1 - 20 of 200 Result page: **1** 2 3 4 5 6 7 8 9 10 next

Best 200 shown

Relevance scale

1 A unified scheduling model for high-level synthesis and code generation

A. Kifli, G. Goosens, H. De Man

March 1995 **Proceedings of the 1995 European conference on Design and Test**

Publisher: IEEE Computer Society

Full text available: pdf(686.11 KB) Publisher Site Additional Information: full citation, abstract, citings

Scheduling is an essential task both in high-level synthesis and in code generation for programmable processors. In this paper we discuss the impact of the controller model on the scheduling task for DSP applications. Existing techniques in high-level synthesis mostly assume a simple controller model in the form of a single FSM. However, in reality more complex controller architectures are often used. On the other hand, in the case of programmable processors, the controller architecture is large ...

Keywords: DSP applications, FSM, application specific integrated circuits, code assembly, code generation, control-flow instructions, controller generation, controller model, digital signal processing chips, finite state machines, high level synthesis, high-level synthesis, machine code description, processor scheduling, programmable processors, real-time systems, scheduling, unified scheduling model

2 Principles of conservative parallel simulation

David M. Nicol

November 1996 **Proceedings of the 28th conference on Winter simulation**

Publisher: ACM Press

Full text available: pdf(893.04 KB) Additional Information: full citation, references, citings

3 Effectivness of abstract interpretation in automatic parallelization: a case study in logic

programming
Francisco Bueno, María García de la Banda, Manuel Hermenegildo
March 1999 **ACM Transactions on Programming Languages and Systems (TOPLAS),** Volume 21 Issue 2
Publisher: ACM Press
Full text available: pdf(533.48 KB) Additional Information: full citation, abstract, references, citings, index terms

We report on a detailed study of the application and effectiveness of program analysis based on abstract interpretation of automatic program parallelization. We study the case of parallelizing logic programs using the notion of strict independence. We first propose and prove correct a methodology for the application in the parallelization task of the information inferred by abstract interpretation, using a parametric domain. The methodology is generic in the sense of allowing the use of dif ...

Keywords: abstract interpretation, automatic parallelization, data flow analysis, logic programming, parallelism

4 Optimization for a superscalar out-of-order machine
Anne M. Holler
December 1996 **Proceedings of the 29th annual ACM/IEEE international symposium on Microarchitecture**
Publisher: IEEE Computer Society
Full text available: pdf(1.55 MB) Additional Information: full citation, abstract, citings, index terms

Compiler optimization plays a key role in unlocking the performance of the PA-8000, an innovative dynamically-scheduled machine which is the first implementation of the 64-bit PA 2.0 member of the HP PA-RISC architecture family. This wide superscalar, long out-of-order machine provides significant execution bandwidth and automatically hides latency at runtime; however, despite its ample hardware resources, many of the optimizing transformations which proved effective for the PA-8000 served to au ...

5 Concurrent garbage collection using hardware-assisted profiling
Timothy H. Heil, James E. Smith
October 2000 **ACM SIGPLAN Notices , Proceedings of the 2nd international symposium on Memory management ISMM '00**, Volume 36 Issue 1
Publisher: ACM Press
Full text available: pdf(1.74 MB) Additional Information: full citation, abstract, citings, index terms

In the presence of on-chip multithreading, a Virtual Machine (VM) implementation can readily take advantage of *service threads* for enhancing performance by performing tasks such as profile collection and analysis, dynamic optimization, and garbage collection concurrently with program execution. In this context, a hardware-assisted profiling mechanism is proposed. The *Relational Profiling Architecture* (RPA) is designed from the top down. RPA is based on a relational model similar ...

6 Avoiding conditional branches by code replication
Frank Mueller, David B. Whalley

June 1995 **ACM SIGPLAN Notices , Proceedings of the ACM SIGPLAN 1995 conference on Programming language design and implementation PLDI '95**, Volume 30 Issue 6
Publisher: ACM Press
Full text available: pdf(1.10 MB) Additional Information: full citation, references, citings, index terms

7 Optimally profiling and tracing programs
Thomas Ball, James R. Larus
July 1994 **ACM Transactions on Programming Languages and Systems (TOPLAS)**, Volume 16 Issue 4
Publisher: ACM Press
Full text available: pdf(2.84 MB) Additional Information: full citation, abstract, references, citings, index terms, review

This paper describes algorithms for inserting monitoring code to profile and trace programs. These algorithms greatly reduce the cost of measuring programs with respect to the commonly used technique of placing code in each basic block. Program profiling counts the number of times each basic block in a program executes. Instruction tracing records the sequence of basic blocks traversed in a program execution. The algorithms optimize the placement of counting/tracing code with respect to the ...

Keywords: control-flow graph, instruction tracing, instrumentation, profiling

8 A modified random perturbation method for database security
Patrick Tendick, Norman Matloff
March 1994 **ACM Transactions on Database Systems (TODS)**, Volume 19 Issue 1
Publisher: ACM Press
Full text available: pdf(1.04 MB) Additional Information: full citation, abstract, references, citings, index terms, review

The random data perturbation (RDP) method of preserving the privacy of individual records in a statistical database is discussed. In particular, it is shown that if confidential attributes are allowed as query-defining variables, severe biases may result in responses to queries. It is also shown that even if query definition through confidential variables is not allowed, biases can still occur in responses to queries such as those involving proportions or counts. In either ...

Keywords: bias, correlation, noise addition, random perturbation method

9 Metamodel estimation using integrated correlation methods
Jeffrey D. Tew, James R. Wilson
December 1987 **Proceedings of the 19th conference on Winter simulation**
Publisher: ACM Press
Full text available: pdf(750.47 KB) Additional Information: full citation, abstract, references, citings, index terms

This paper develops a generalized approach for combining the use of the Schruben-Margolin correlation induction strategy and control variates in a simulation experiment designed to estimate a metamodel that is linear in the unknown parameters relating the response variable of interest to selected exogenous decision variables. This generalized approach is based on standard techniques of regression analysis. Under certain broad assumptions, the combined use of the Schruben-Margolin correlatio ...

10 Static slicing in the presence of goto statements
Jong-Deok Choi, Jeanne Ferrante
July 1994 **ACM Transactions on Programming Languages and Systems (TOPLAS)**, Volume 16 Issue 4
Publisher: ACM Press
Full text available: pdf(1.00 MB) Additional Information: full citation, abstract, references, citings, index terms, review

A static program slice is an extract of a program which can help our understanding of the behavior of the program; it has been proposed for use in debugging, optimization, parallelization, and integration of programs. This article considers two types of static slices: executable and nonexecutable. Efficient and well-founded methods have been developed to construct executable slices for programs without goto statements; it would be tempting to assume these methods would apply as well in prog ...

Keywords: debugging, program analysis, slicing, testing

11 Logic and logic programming
J. A. Robinson
March 1992 **Communications of the ACM**, Volume 35 Issue 3
Publisher: ACM Press
Full text available: pdf(6.56 MB) Additional Information: full citation, references, citings, index terms

Keywords: unification

12 Analysis of the conditional skip instructions of the HP precision architecture
Jonathan P. Vogel, Bruce K. Holmer
November 1994 **Proceedings of the 27th annual international symposium on Microarchitecture**
Publisher: ACM Press
Full text available: pdf(848.20 KB) Additional Information: full citation, abstract, references, index terms

The HP-PA instruction set allows any arithmetic instruction to conditionally skip the following instruction based on the result of the arithmetic calculation. We have isolated this architectural feature and measured its performance benefit on a set of SPEC benchmark programs. Results indicate that adding the ability to skip to arithmetic instructions yields only a marginal performance benefit (less than 0.3%) for floating point intensive programs. For integer programs, however, the average ...

13 Reports from related meetings: Interface '99: a data mining overview
Arnold Goodman

January 2000 **ACM SIGKDD Explorations Newsletter**, Volume 1 Issue 2
Publisher: ACM Press
Full text available: pdf(851.62 KB) Additional Information: full citation, abstract, references

This personal overview of Interface '99 is intended to communicate its meaning and relevance to SIGKDD, as well as provide valuable information on trends within the Interface for data miners seeking to learn more about statistics. In addition, it is the newest link in a bridge between the Interface and KDD begun by References 2-4 and the sessions on KDD at Interface '98 and Interface '99.

Keywords: review of Interface'99 conference, statistics

14 Formally based static analysis of microcode
J. M. Foster
December 1986 **ACM SIGMICRO Newsletter , Proceedings of the 19th annual workshop on Microprogramming MICRO 19**, Volume 17 Issue 4
Publisher: ACM Press
Full text available: pdf(822.75 KB) Additional Information: full citation, abstract, references, index terms

Algebraic methods have been widely used to find properties of programs, especially for use in compiler optimisation. This paper describes the use of this kind of method to prove the absence of particular errors in microcode, or to detect and locate such errors. In order to show the kind of error which may be found we consider a number of examples. All of these have found errors in practical microcode, written for the PERQ computer.

15 Conditional attribute grammars
John Tang Boyland
January 1996 **ACM Transactions on Programming Languages and Systems (TOPLAS)**, Volume 18 Issue 1
Publisher: ACM Press
Full text available: pdf(222.04 KB) Additional Information: full citation, abstract, references, citings, index terms

Attribute grammars are a useful formalism for the specification of computations on structured terms. The classical definition of attribute grammars, however, has no way of treating conditionals nonstrictly. Consequently, the natural way of expressing many otherwise well-behaved computations involves a circularity. This article presents conditional attribute grammars, and extension of attribute grammars that enables more precise analysis of conditionals. In conditional attri ...

Keywords: attribute grammars, conditionals, demand evaluation, functional dependencies, language processor generators, nonstrict evaluation, static analysis

16 Simplifying termination proofs for rewrite systems by preprocessing
Bernhard Gramlich
September 2000 **Proceedings of the 2nd ACM SIGPLAN international conference on Principles and practice of declarative programming**

Publisher: ACM Press
Full text available: pdf(331.77 KB) Additional Information: full citation, references, index terms

17 An algorithm for optimal lambda calculus reduction
John Lamping
December 1989 **Proceedings of the 17th ACM SIGPLAN-SIGACT symposium on Principles ol programming languages**
Publisher: ACM Press
Full text available: pdf(1.56 MB) Additional Information: full citation, abstract, references, citings, index terms

We present an algorithm for lambda expression reduction that avoids any copying that could later cause duplication of work. It is optimal in the sense defined by Lévy. The basis of the algorithm is a graphical representation of the kinds of commonality that can arise from substitutions; the idea can be adapted to represent other kinds of expressions besides lambda expressions. The algorithm is also well suited to parallel implementations, consisting of a fixed set of local graph rewr ...

18 Compiler techniques for code compaction
Saumya K. Debray, William Evans, Robert Muth, Bjorn De Sutter
March 2000 **ACM Transactions on Programming Languages and Systems (TOPLAS),** Volume 22 Issue 2
Publisher: ACM Press
Full text available: pdf(409.20 KB) Additional Information: full citation, abstract, references, citings, index terms, review

In recent years there has been an increasing trend toward the incorpor ation of computers into a variety of devices where the amount of memory available is limited. This makes it desirable to try to reduce the size of applications where possible. This article explores the use of compiler techniques to accomplish code compaction to yield smaller executables. The main contribution of this article is to show that careful, aggressive, interprocedural optimization, together with procedural abstr ...

Keywords: code compaction, code compression, code size reduction

19 On randomization in sequential and distributed algorithms
Rajiv Gupta, Scott A. Smolka, Shaji Bhaskar
March 1994 **ACM Computing Surveys (CSUR),** Volume 26 Issue 1
Publisher: ACM Press
Full text available: pdf(8.01 MB) Additional Information: full citation, abstract, references, citings, index terms

Probabilistic, or randomized, algorithms are fast becoming as commonplace as conventional deterministic algorithms. This survey presents five techniques that have been widely used in the design of randomized algorithms. These techniques are illustrated using 12 randomized algorithms—both sequential and distributed— that span a wide range of applications, including:primality testing (a classical problem in number theory), interactive probabilistic proof s ...

Keywords: Byzantine agreement, CSP, analysis of algorithms, computational complexity, dining philosophers problem, distributed algorithms, graph isomorphism, hashing, interactive probabilistic proof systems, leader election, message routing, nearest-neighbors problem, perfect hashing, primality testing, probabilistic techniques, randomized or probabilistic algorithms, randomized quicksort, sequential algorithms, transitive tournaments, universal hashing

20 Performance evaluation of instruction scheduling on the IBM RISC System/6000
David Bernstein, Doron Cohen, Yuval Lavon, Vladimir Rainish
December 1992 **ACM SIGMICRO Newsletter , Proceedings of the 25th annual international symposium on Microarchitecture MICRO 25**, Volume 23 Issue 1-2
Publisher: IEEE Computer Society Press, ACM Press
Full text available: pdf(1.13 MB) Additional Information: full citation, references, citings, index terms

Results 1 - 20 of 200 Result page: **1** 2 3 4 5 6 7 8 9 10 next

Terms of Usage Privacy Policy Code of Ethics Contact Us

Useful downloads: Adobe Acrobat QuickTime Windows Media Player Real Playe

IEEE Xplore® RELEASE 2.1

Home | Login | Logout | Access Informatio Siter

Welcome United States Patent and Trademark Office

Search Results

BROWSE SEARCH IEEE XPLORE GUIDE

Results for "(((yield points sampling instrumentation)<in>metadata)) <and> (pyr >= 1985 <and>..."

✉e-mail

Your search matched **0** documents.
A maximum of **100** results are displayed, **25** to a page, sorted by **Relevance** in **Descending** order.

» Search Options

View Session History

New Search

Modify Search

(((yield points sampling instrumentation)<in>metadata)) <and> (pyr >= 1!

□ Check to search only within this results set

Display Format: ● Citation ○ Citation & Abstract

» Key

IEEE JNL	IEEE Journal or Magazine
IEE JNL	IEE Journal or Magazine
IEEE CNF	IEEE Conference Proceeding
IEE CNF	IEE Conference Proceeding
IEEE STD	IEEE Standard

No results were found.

Please edit your search criteria and try again. Refer to the Help pages assistance revising your search.

Help Contact U Securi

indexed by inspec®

Home | Login | Logout | Access Informatio
Siter

Welcome United States Patent and Trademark Office

Search Results

BROWSE SEARCH IEEE XPLORE GUIDE

Results for **"(((yield points sampl* instrument*)<in>metadata)) <and> (pyr >= 1985 <and> pyr ..."** e-mail
Your search matched **0** documents.
A maximum of **100** results are displayed, **25** to a page, sorted by **Relevance** in **Descending** order.

» Search Options

View Session History

New Search

Modify Search

(((yield points sampl* instrument*)<in>metadata)) <and> (pyr >= 1985 <a

□ Check to search only within this results set

Display Format: ● Citation ○ Citation & Abstract

» Key

IEEE JNL	IEEE Journal or Magazine
IEE JNL	IEE Journal or Magazine
IEEE CNF	IEEE Conference Proceeding
IEE CNF	IEE Conference Proceeding
IEEE STD	IEEE Standard

No results were found.

Please edit your search criteria and try again. Refer to the Help pages assistance revising your search.

Help Contact I
Securi

Indexed by
Inspec

Home | Login | Logout | Access Informatio
Siter

IEEE Xplore®
RELEASE 2.1

Welcome United States Patent and Trademark Office

Search Results

BROWSE SEARCH IEEE XPLORE GUIDE

Results for **"(((conditional* unconditional* sampl*)<in>metadata)) <and> (pyr >= 1985 <and> ..."** e-mail
Your search matched **0** documents.
A maximum of **100** results are displayed, **25** to a page, sorted by **Relevance** in **Descending** order.

» Search Options

View Session History

New Search

Modify Search

(((conditional* unconditional* sampl*)<in>metadata)) <and> (pyr >= 1985

□ Check to search only within this results set

Display Format: ● Citation ○ Citation & Abstract

» Key

IEEE JNL	IEEE Journal or Magazine
IEE JNL	IEE Journal or Magazine
IEEE CNF	IEEE Conference Proceeding
IEE CNF	IEE Conference Proceeding
IEEE STD	IEEE Standard

No results were found.

Please edit your search criteria and try again. Refer to the Help pages assistance revising your search.

Help Contact U
Securi

indexed by
inspec®